Days off scheduling

A 2-phase approach to personnel rostering

Sophie van Veldhoven · **Gerhard Post** · **Egbert van der Veen**

1 Introduction

The personnel rostering problem (or nurse rostering problem) is a well-known combinatorial optimization problem, with a rich literature, see [Ernst et al(2004)] and [Burke et al(2004)]. In many practical situations the large number of working time regulations and preferences make it difficult and time consuming to construct a good schedule.

One way to reduce the complexity of the personnel rostering problem is to decompose the problem into subproblems that are easier to solve. Though the subproblems can possibly be solved to optimality in a reasonable time, the combination of the subproblems does not necessarily lead to an optimal solution to the original problem. In this work we present a 2-phase decomposition model. In the first phase we construct a *days off schedule*, that is a schedule that specifies for each employee the working days and the days off. We also consider a refinement in which we consider day off, day shift or night shift. In the second phase we specify which shifts are actually assigned to the employees on their working days, which means that we solve a *shift rostering problem* respecting the days off schedule found in the first phase.

In practice, the construction of a schedule with working days and days off can be a separate step in the process of assigning staff. For individual employees, it may be pleasant to know the working days a long time ahead, so that they can plan their free

Sophie van Veldhoven
Aviv, Langestraat 11, 7511 HA Enschede, The Netherlands,
E-mail: sophievveldhoven@gmail.com

Gerhard Post (corresponding author)
Department of Applied Mathematics, University of Twente, P.O. Box 217, Enschede, The Netherlands,
and ORTEC, Groningenweg 6k, 2803 PV Gouda, The Netherlands,
E-mail: g.f.post@utwente.nl

Egbert van der Veen
Department of Applied Mathematics, University of Twente, P.O. Box 217, Enschede, The Netherlands,
and ORTEC, Groningenweg 6k, 2803 PV Gouda, The Netherlands,
E-mail: Egbert.vanderVeen@ortec.com

time. The exact hours they have to work, are not essential to know in the long term. In addition, requests for vacation can be taken into account long before the actual planning, which can alert the planner for possible capacity problems. The assignment of the various shifts to the employees can be done in a later stage, for example, a month before the start of the new planning period.

The aim of this paper is two-fold: investigate the computation times for a class of personnel rostering problems with and without decomposition, and, secondly, investigate the quality of the schedules in the 2-phase approach compared to solutions obtained in one run. We will use integer linear programming (ILP) in our research. Hence we will present a mathematical program for the construction of days off schedules, based on the constraints of the original problem.

2 Approach

As data sets we use the Employee Scheduling Benchmark Data Sets [Curtois(2007)]. We formulate formulate an ILP for these data sets, which are solved using CPLEX 12.2, with the time limit set to 1 hour. These results serve as the benchmark for our tests. Next we formulate the days off scheduling problems, which we derive from the instances. We consider two variants:

(On, Off): Distinguish working days and days off.
(Day, Night, Off): Distinguish day shifts, night shifts, and days off.

We include the refinement (Day, Night, Off) in our study, because in many cases the requirements before and after night shifts are very different from (all) other shifts. In fact we reduce the original shift scheduling problem to a shift scheduling problem with 2 shift types (on, off), respectively 3 shift types (day, night, off). Hence in the first and second phase we can use similar ILP models, with the addition that in the second phase the assignment of shifts to employees should obey the decisions of the first phase. Though the ILP models are similar, the reduction in the first phase is not straightforward, and the next subsections describe how this is done. For this, we consider the different types of constraints present in the benchmark, and explain shortly how we handle them.

Requests and fixed assignments

Employees can have shifts preassigned, can have requests for shifts on or can have requests for a day off. Those can be transferred to the days off scheduling problem directly: work requests or shift on requests result in days on. Shift *off* requests are ignored, since the employee might work on the same day in another shift.

Cover requirements

Per day the instance describes a minimum, maximum, or preferred cover per shift type or per time period. These have to be aggregated to be useful in the days off schedule. In case the cover requirements are in (overlapping) shift types we formulate an ILP model to determine the minimum number of employees needed, and use this amount for the days off schedule. In case that time periods are used, we proceed in the same way.

Pattern requirements

The pattern requirements can express a wide variety of constraints for individual schedules. It can contain constraints like the maximum number of shifts per planning interval or week, shift sequences, shifts in weekends, etc. If a pattern concerns all shift types, we can use it in the days off schedule. In the case of (day, night, off), we can incorporate more pattern requirements, potentially leading to better results.

Workload requirements

Workload requirements describe the number of hours an employee should work in the planning period. Clearly, these are difficult to use in the days off schedule if different shift types have different lengths. Since we use only necessary conditions, the best we can do is to calculate upper and lower bounds on the working days, and add those conditions as constraints to the ILP. In the shift scheduling phase we have full information, so that we can use the correct workload requirements.

3 Results

We tested the 2-phase decomposition on 16 instances present at [Curtois(2007)] with 3 to 12 shift types. For almost all of these instances optimal solutions are known. Running CPLEX 12.2 for 1 hour on a standard dual core PC yields an optimal solution in 9 cases. For these cases we find that the (on, off) decomposition gives only good results for 4 of these instances, where using the (day, night, off) decomposition leads to good results for 6 instances. For the other 7 cases we compare the decomposition results with the results without decomposition. Then we find good results in 3 out of 7 cases if we use the (on, off) decomposition, and in 5 cases if we use the (day, night, off) decomposition. In most cases the calculation times reduce by more than 98%. Here 'good results' is meant in the sense that the cost is of the same order of magnitude; sometimes the results are better, but usually slightly worse.

Several improvements can be made to the model and the solution method. For the model we can improve the way that the pattern constraints are dealt with; now in several instances we use only a minor portion of these in the days off schedule, which in some instances leads to high costs in phase 2. For the solution part, we can apply the decomposition several times, say 10 times, with a constraint added that makes sure we do not generate the same solution. On some instances, this approach significantly improves the results.

References

[Burke et al(2004)] Burke EK, De Causmaecker P, Vanden Berghe G, Van Landeghem H (2004) The state of the art of nurse rostering. Journal of Scheduling 7(6): pp. 441–499.

[Curtois(2007)] Curtois T (2007) `http://www.cs.nott.ac.uk/∼tec/NRP`, Employee Scheduling Benchmark Data Sets.

[Ernst et al(2004)] Ernst AT, Jiang H, Krishnamoorthy M, Owens B, Sier D (2004) An annotated bibliography of personnel scheduling and rostering. Annals of Operations Research 127: pp. 21–144.